

Research and Application on validation.xml of Struts2 Framework Verify File

Fenghua Li

Bohai University, Jinzhou, P.R. China

lzy8917891@163.com

Keywords: Struts2; framework; validation file; validation.xml

Abstract. Input validation is the problem that need to solve for all Web applications, to calibrate data for its constraints and rules. The Struts2 framework provides a variety of check method, and validation.xml file is the most common method in this research. Firstly, study Struts2 framework architecture composed of three parts such as model, view and controller; Then, show the input checking process of Struts2 through a graphical mode ;Finally, the design of validation.xml validation file example, realizes the checkers such as required string, string length, integer, date and double precision floating point number. Using Validation.xml authentication of Struts2 has the advantages of convenient and suitable, and can also be combined with other methods in practical application.

Introduction

Struts2 is a product of the next generation of Struts, and the merger of the new framework in technology based on struts1 and WebWork. Struts2 takes WebWork as the core, using interceptor mechanism to handle the user's request, make the business logic controller can complete off the and ServletAPI [1,2]. Struts2 has the advantages of providing a large number of interceptors, plug-in framework, a variety of views support, more modular, integrated with Spring, easy to test based on the POJO, so it has been used more and more widely. Because of Struts2's powerful function and advantage, it is obtained the widespread application in software development.

Because of the opening of Web application, all Internet browsing can use the application, so the application collecting data through the input page is very complicated, not only contains the normal users' input error, also may contain malicious users' malicious input. A robust application system must prevent illegal input block outside the application and prevent these illegal input into the system, so as to ensure the system is not affected. Therefore, Input validation is the problem that need to solve for all Web applications [3]. Use of the validation.xml file is the most basic input validation provided by Struts2. Struts2 individually appoint a validation file for each Action named <Action-name>-validation.xml which is stored in the same package to the Action, and the check can be realized only to configure a validation file. Elsewhere in the system does not need to change, the system will automatically load the file. This article launches the research on the relevant problems of validation.xml validation file.

Architecture of Struts2 Framework

The MVC model is proposed by Trygve Reenskaug which is a software design pattern for programming language Smalltalk invented by Xerox PARC. MVC mode is to achieve the purpose of the program design of a dynamic, so that simplify the modification and expansion to the program subsequently, and the reuse of a part of the procedures is possible, through the simplification to the complexity, make the program structure more intuitive. Struts2 is a MVC framework realized based on the JSP Model, mainly divided into three parts such as Model, Viewer and Controller, the design concept is decoupling show logic and business logic by the controller, in order to improve maintainability, scalability and reusability of the system [4]. The system structure of the Struts framework is shown in Fig. 1 as follow.

(1) Model. The use is the data related to business logic for packaging and application and

processing method of data. Struts does not define the specific implementation of Model layer, Model layer is usually closely associated with the business logic and it has a continuous requirement. At present, in the field of commercial and open source world, there are some good tools can provide convenience for the development of Model layer.

(2) Viewer. It's the part to process data display in applications. The view is mainly composed of JSP pages which has no process logic, business logic and model information but only marking. Struts itself contains a set of TagLib, which is one of the essence of Struts, flexible application can simplify the code on JSP page and improve development efficiency.

(3) Controller. It has played an Organized role among different layers which is used to control the flow of the application. Controller of Struts is mainly ActionServlet provided by its own. ActionServlet receives all requests from clients and according to the definition in the configuration file (struts-config.xml) transfers control to the appropriate Action object.

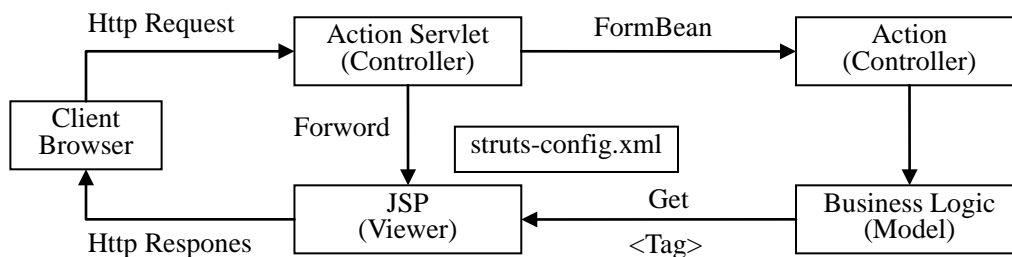


Fig. 1. Architecture of Struts2 Framework

Input Validation Process of Struts2

Input validation process of Struts2 is shown as Fig. 2.

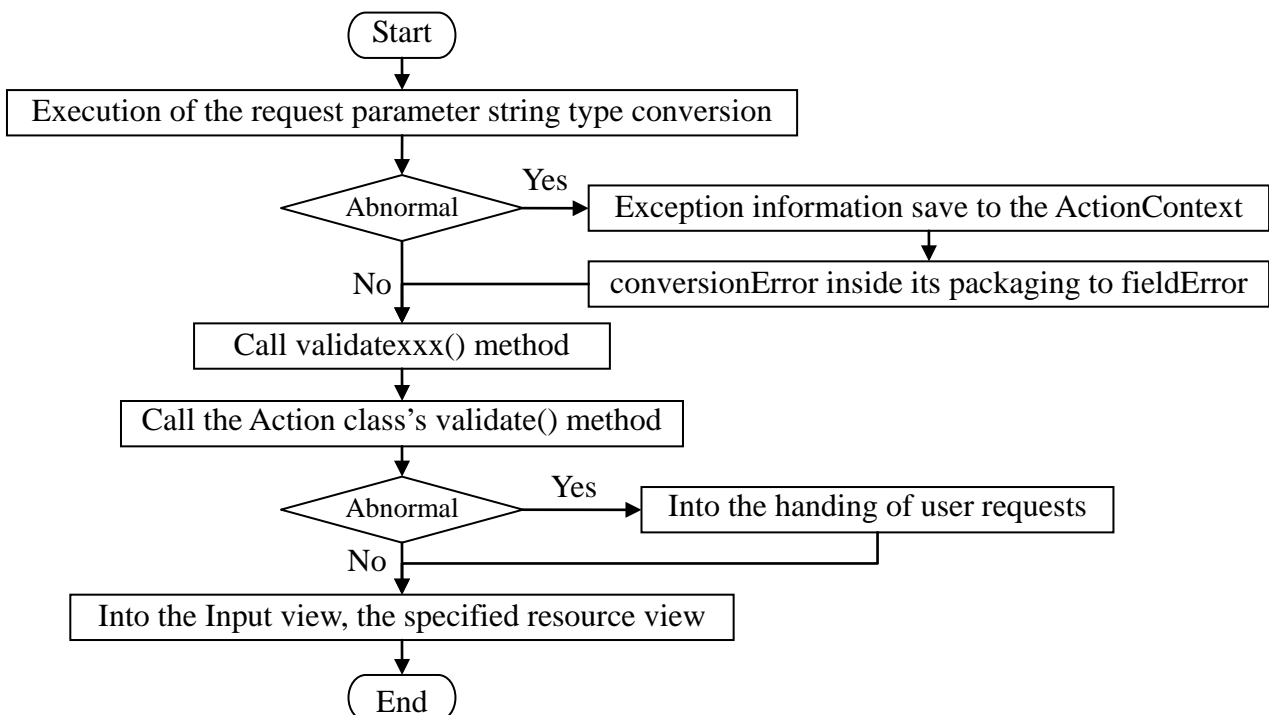


Fig. 2. Input Inspection Process of Struts2

According to Fig. 2, input validation of Struts2 is usually through the following steps [5]:

(1) A type executive type conversion to the request parameter of string, and set these values to the value of the Action attribute.

(2) If the type conversion process is abnormal, the abnormal information saved to the ActionContext, the conversionError interceptor will package it into the fieldError, and executive the

next step. If it has no abnormal, executive the next step directly.

(3) Call the method validateXxx () by reflected, and the Xxx is the name of the method corresponded to the process logic of the user request.

(4) Calls the validate () method in the Action class.

(5) If an exception occurs, directly into the specified view resources of the input view (for the verification of the method must be specified for the input in the configuration file). If no abnormal, then transferred to the processing method of user request, and then transferred to the specified view resources of the input view.

Verify File Design Example of Validation.xml

Taking college English grade examination management table as an example, to illustrate the design method of validation.xml files, the data structure is shown as table 1. In order to simplify the design, Table 1 contains only 6 fields.

Table 1. Score management college English test

No	Name	Fields Type	Fields Width	Fields description
0	ID	Decimal	10,0	Automatic increase
1	ticket	Varchar	10	Ticket number
2	name	Varchar	50	Student name
3	age	Varchar	50	Student age
4	edate	Datetime	40	Examination date
5	score	Decimal	5,2	English exam score

For 6 fields in Table 1, the "ID" field grow automatically, need not verify, the "ticket" field must be filled, "name" has a length limit, the "age" field has a size limit, the "edate" field has interval constraints, the "score" field has a decimal. The validation file are as follows:

<pre><?xml version="1.0" encoding="UTF-8" ?> <!DOCTYPE validators PUBLIC "-//OpenSymphony Group//XWork Validator 1.0/EN" http://www.opensymphony.com/xwork/xwork-v alidator-1.0.2.dtd" > <validators > <field name="ticket"> <!--Check examinalational certification number--> <field-validator type="requiredstring"> <param name="trim">true</param> <message> Examinalational certification number must be filled</message> </field-validator> </field> <field name="name"> <!-- Check student's name --> <field-validator type="stringlength"> <param name="minlength">2</param> <param name="maxlength">30</param> <message> The length of student's name must be from 2 to 30</message> </field-validator> </field> <field name="age"> <!-- Check age --></pre>	<pre><field-validator type="int"> <param name="min">10</param> <param name="max">100</param> <message> The age must between 10 and 100</message> </field-validator> </field> <field name=" edate"> <!-- Check examination date --> <field-validator type="date"> <param name="min">2000-01-01</param> <param name="max">2050-12-31</param> <message> The examination dates must between \${min} and \${max}</message> </field-validator> </field> <field name="score"> <!-- Check examination score --> <field-validator type="double"> <param name="min">0</param> <param name="max">999.99</param> <message> The examination score must between 0 and 999.99</message> </field-validator> </field> </validators></pre>
---	--

Xwork-2.0.4.jar package (com.opensymphony.xwork2.validator.validators) has a default.xml file which defines the checker in the construction of Struts2 framework. The default.xml file defines the

commonly used checker types. The validation file above relates to 5 kinds of checkers, they are described as follows [6]: (1) Required string checker, the requirements of field value must be a non empty with length greater than 0; (2) The string length checker, asked the length be checked of the field must be in the specified range, otherwise check failed; (3) Integer checker, asked the integer value must be in the specified range, and can also not check the minimum or maximum value; (4) Date checker, to check whether a date in the specified range, if not in the specified range, it will generate an error message; (5) Double precision floating point number checker, which similar to the integer checker, used to retrieve field values range of double precision floating-point number.

Conclusion

For Web applications, all users are collecting data through the browser, if the operation of the user is not skilled, it will input error, hardware equipment is abnormal or network transmission is not stable, even malicious user's sabotage, these may cause abnormal input and cause non normal interruption the system in light, or lead to the breakdown of the system in heavy [7]. Data validation is an important function need to achieve in the development of software, to calibrate data for its constraints and rules, and filtered data not meet the requirements, so as to improve the data effectiveness of system [8]. The validation.xml of the Struts2 studied in this paper has the advantages of convenient and suitable. But the actual application needs to pay attention to the following questions: to the Action class name with "-validation.xml" as the name of the file; the LoginAction-validation.xml file must be associated with the class Action in the same directory; the type attribute of "<field-validator type=" > tag in the validation.xml file, must be the same with the data type defined in the VO .

Acknowledgement

This work is supported by "Application of the Open Courses in College English Viewing Listening and Speaking", a general subject of "the Twelfth Five-year Plan" on Educational Science of Liaoning Province, 2012, Approval (JG12DB213).

References

- [1] Z. G. Zhang, "Development and application on Struts2 framework technology," Computer CD Software and Applications, vol. 16, no. 23, pp. 119-120, 2013.
- [2] Baidu Baike, "Struts2," http://baike.baidu.com/view/1566725.htm?from_id=10711345&type=syn&fromtitle=Struts2&fr=aladdin, 2014-11-21.
- [3] G. Li, "Struts2 Definitive Guide," Publishing House of Electronic Industry, 2008.
- [4] Y. B. Guo, S. M. Lu, "Universal design based on Struts2 framework," Computer Programming Skills & Maintenance, vol. 21, no. 11, pp. 21-23, 2014.
- [5] H. Tian, L. Du, "Input Validation Summing-up of the Struts2," <http://zhaohe162.blog.163.com/blog/static/38216797201045115612836/>, 2014-10-21.
- [6] X. M. Li, "Based on the Input Validation Functions Struts2 Framework Implementation and Application," Computer Knowledge and Technology, vol. 9, no. 35, pp. 8201-8202, 2013.
- [7] X. L. Qi, T. S. Guo, Z. Y. Zou, "Analysis and application of data validation based on struts framework," Electronic Instrumentation Customers, vol. 18, no. 5, pp. 48-50, 2011.
- [8] D. C. Zhou, F. J. Jv, D.H. Guo, "Implementation of validation data and integration regular expression based on Struts2 framework," Computer and Modernization, vol. 27, no. 1, pp. 85-87, 2011.